"Data Points"- A new proposed software cost estimation metric to estimate software cost which contain huge database

Mrs. Shefali Trushit Naik

Abstract— In current era, many software cost estimation metrics are available, such as SLOC, Function Points, Feature Points and Object Points. There are various metrics available to estimate cost of Management Information System, Embedded System, Object-Oriented Systems, etc. The parameters to calculate function points are based mainly on complexity of logic written, no. of inputs, no. of outputs, no. of files and interfaces used and many more. But somehow, the database which includes tables, constraints, complexity of data types, retrieval method, etc. are not being considered to estimate software cost. In this paper, the new metric based on the previous metrics is presented by considering some additional factors. The metric is given a name "Data Points" which must be considered to count cost of systems with small to huge databases. Thus, Data Points should also be considered along with Function Points to calculate the cost of software. Recently, there are many e-governance projects has been developed, which contains rigorous amount of data. We need powerful software which should have proper database to manage these data. To estimate software cost of these types of projects, the Data Point Count (DPC) should be considered. I have tried to present this new concept in this paper.

Index Terms— Database, Data Points, Data Point Count, Function Point, Software Cost Estimation, Software Cost Estimation Metric, Software Engineering, Software Estimation.

———————————— ◆ ————————————

1 Introduction

Information and Communication Technology is emerging very fast nowadays. The people who are working in IT industry have also experienced this by changing demands of customers. If you have learnt new technology, it becomes older before you actually use it. Means before we learn something, it gets outdated! There are so many applications of ICT which are used by common man and still they need many more.

For ex., all e-governance applications, mobile applications, cloud applications, knowledge based or expert systems, embedded systems, web applications, simple desktop applications, scientific applications, business applications. The list is endless.

To serve common man's demand, we have thousands of companies who develop these kinds of applications. But very few companies really serve these. We can count those companies on our finger tips. Why this is the scenario? Because all of them are lacking somewhere. Sometimes the budget overruns, planning is not proper, incomplete requirement specification, improper work schedule, resources are not used optimally, poor quality, etc. One major problem with software development is cost-benefit analysis [12]. There are no perfect methods for cost estimation of a software. Because software is not manufactured with raw materials. So we really are not able to find the actual cost. But many software cost estimation metrics are available, which helps to measure software cost very efficiently. These metrics [17] are mainly based on software size i.e. counting lines of code[4, 5, 6, 7 and 8] written in the program, functionalities used in software i.e. counting function points[8, 9 and 10] or feature points for embedded systems, etc. We are following these standard methods to estimate the software cost [1, 2, 3, 11, and 16]. Currently, the database size is not considered to calculate software cost. Here, I have presented a new metric named "Data Points" which is related with counting database functionalities, which would be helpful to decide cost of a software which contains very huge database.

————————————————

- *Mrs. Shefali Trushit Naik is working as a senior lecturer in H.L. Institute of Computer Applications, Ahmedabad University, Ahmedabad, Gujarat, India. She has more than twelve years of experience in academics. Her areas of interest are Databases, Software Engineering, Computer System Architecture, Software Project Management and Programming. E-mail: shefali.naik@ahduni.edu.in*

2 Why the name "Data Point"?

There are always two parts of any software. One part is stored data and another part is processing. The application whether it is big or small, contains some data. In previously developed cost estimation metrics [[4, 5 and 6], data is considered; but not in much details. The focus was on coding lines. But here, exclusively, the focus is not estimation of data and things related to data. Therefore, it has a name "Data Point". Fig 1 shows how software cost should be calculated.

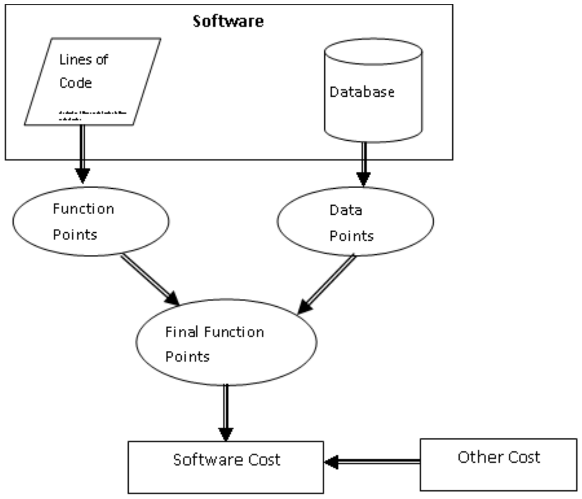

Fig 1 Software Cost Estimation using Final Function Points

3 Importance of Data Point

Currently, there are so many applications which contain huge data storage. For ex., all e-governance applications in India. Some of them are ICT enabled, but many of them are still in process. All these applications contain complex data such as data related to citizen's personal property, unique identification, vehicle registration, health care, police records, court records, birth and death details, passport details, etc. These require storage of biometric information, images, documents scanning and storage, audio and video file storage and other complex details. In these type of applications, processing is important, but management, updation and retrieval of data is more important. We have to decide the type of database which should be used to store these data, security features to be applied as these data are confidential, algorithms for efficient retrieval, and how we can improve performance and many more. If we consider all these aspects and meet all these constraints then it will surely affect the software cost. Hence, "Data Points" count is one of the important metrics for software cost estimation.

4 Counting "Data point"

We can separately calculate data points for the software from the following parameters.

1. Volume of data. The actual average no. of transactions to be processed per day by our software.
2. Type of database management system which is used to manage data. Ex., RDBMS, ORDBMS, No SQL Database, etc. To decide weightage of database, we may consider various factors such as security features provided, support for PL/SQL, Transaction management capacity, Recovery management, Storage Capacity, etc.[18]
3. Environment in which data will be accessed. Ex., Single-user, Multi-user, Web-based, Client-Server, etc.
4. Total no. of Tables category wise. For ex., Simple tables, Partitioned tables, Nested tables, etc.
5. Complexity of Keys. Complexity of keys can be decided from no. of fields contained in the primary keys or foreign keys.
6. Complexity of Relationships
7. No. of Objects other than Tables. Ex., Stored Procedures, Stored Functions, Triggers, Methods, etc. For these types of objects, Total Lines of Codes can be calculated or Function points can be calculated by using any standard method such as Albrecht FP [8, 9, and 10], COCOMO [13, 14 and 15] or any other method.
8. Total no. of Indexes. Complexity of indexes depends on total no. of fields used to create an index.
9. Volume of total no. of daily transactions handled by the database.
10. Total no. of complex fields. For ex., Fields to store images, documents, derived fields, audio and video files, etc.

Table 1 can be used to decide the weightage of each component of database according to their complexity and volume. The table may be considered as a base. But after all it depends on the organization to decide the weightages, because the company who is actually developing the software knows better about the cost of all components used in the system development.

From Table 1, the Data Points can be calculated for the software as following.

Data Point Count (DPC) =sum of (Weightage of each parameter*Volume of each parameter), Where there is no volume given, we have to consider 1 as multiplier.

$$\text{Therefore, DPC} = \sum_{i=1}^{n} (W_i * V_i)$$

Where W_i=Weightage of each parameter
V_i = Volume of each parameter
n=Total no. of parameters

After calculating data point counts, it can be added in Function Point Count to estimate the final cost of software.

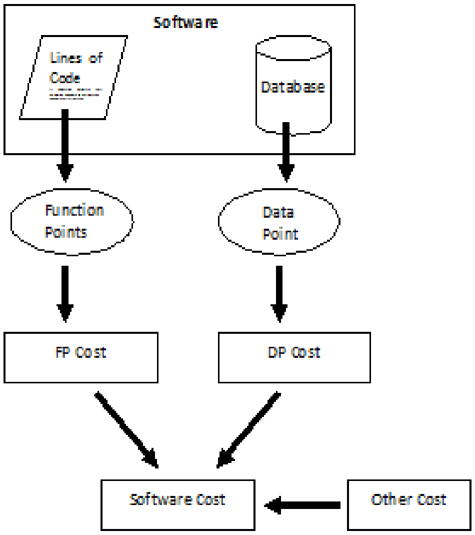

Fig 2 Software Cost Estimation using independent cost of Function Points and Data Points

We may also calculate, Data Point cost and Function Point cost independently. The final cost can be derived by adding other costs into these two. Fig 2 shows the alternative way to calculate software's final cost.

After finding final function point count or data point count, we may decide the cost according to some standard cost table. The cost may vary from organization to organization, because it may be affected by experience of staff involved in software development, company policies, success rates of previously developed software, resources used to develop software and many more.

REFERENCES

[1] V. Bozhikova, M. Stoeva, "An Approach for Software Cost Estimation" , *International Conference on Computer Systems and Technologies-CompSysTech'10*

[2] S. Grimstad and M. Jorgensen, "Framework for the Analysis of Software Cost Estimation Accuracy", *ISESE '06*, September 21-22, 2006, Rio de Janeiro, Brazil

[3] N. Mittas and L. Angelis, "Combining Regression and Estimation by Analogy in a Semi-parametric Model for Software Cost Estimation", *ESEM '08*, October 9-10, 2008, Kaiserslautern, Germany

[4] M. Burger and A. Zeller, "Minimizing Reproduction of Software Failures", *ISSTA '11*, July 17-21, 2011, Toronto, ON, Canada

[5] V. Nguyen, L. huang and B. Boehm, "An Analysis of Trends in Productivity and Cost Drivers Over Years", *PROMISE '11*, September 20-21, 2011, Banff, Canada

[6] B. Vasilescu, A. Serebrenik and M. Brand, "By No Means : A Study on Aggregating Software Metrics", *WETSoM*, May 24, 2011, Waikiki, Honolulu, HI, USA

[7] K. Lind and R. Heldal, "On the Relationship Between Functional Size and Software Code Size", *WETSoM '10*, May 4, 2010, Cape Town, South Africa

[8] P. Ferreira and M. Barros, "Traceability Between Function Point and Source Code", *TEFSE '11*, May 23, 2011, Honolulu, HI, USA

[9] L. Lavazza and C. Garavaglia, "Using Function Points to Measure and Estimate Real-Time and Embedded Software : Experiences and Guidelines", *Third International Symposium on Empirical Software Engineering and Measurement"*

[10] Dr. V. Saxena and M. Shrivastava, "Performance of Function Point Analysis Through UML Modeling", *SIGSOFT Software Engineering Notes*, March 2009, Volume 34, Number 2

[11] P. Keil, D. Paulish and R. Sangwan, "Cost Estimation for Global Software Development", *EDSER '06*, May 27, 2006, Shanghai, Chaina

[12] I. Learman, "Cost Benefits Analysis in the Design and Evaluation of Information Systems", *Fall Joint Conference*, 1971

[13] C. Garcia and C. Hirata, "Integrating Functional Metrics, COCOMO II and Earned Value Analysis for Software Projects using PMBoK", *SAC '08*, March 16-20, 2008, Fortaleza, Ceara, Brazil

[14] Y. Zheng, K. Wang and J. Xue, "An Extension of COCOMO II for the B-Method", *EDSER '06*, May 27, 2006, Shanghai, Chaina

[15] W. Jiamthubthugsin and D. Sutivong, "Portfolio Management of Software Development Projects using COCOMO II", *ICSE '06*, May 20-28, 2006, Shanghai, Chaina

[16] M. Ayyildiz, O. Kalipsiz and S. Yavuz, " A Metric-Set and Model Suggestion for Better Software Project Cost Estimation", *World Academy of Science, Engineering and Technology 23 2006*

[17] Book : E. Mills, "Software Metrics", *SEI Curriculum Module SEI-CM-12-1.1*, December, 1988

[18] Book : C. J. Date, "An Introduction to Database Systems", Seventh Edition, Pearson Education Asia

About the Author

Mrs. Shefali Trushit Naik is working as a Senior Lecturer in H.L. Institute of Computer Applications, Ahmedabad University, Ahmedabad, Gujarat, India. She has more than twelve years of experience in academics. Her areas of interest are Databases, Software Engineering, Computer System Architecture, Software Project Management and Programming. E-mail: shefali.naik@ahduni.edu.in